COP 4610L: Applications in the Enterprise
Spring 2005

Introduction to PHP – Part 3

Instructor: Mark Llewellyn
markl@cs.ucf.edu
CSB 242, 823-2790

http://www.cs.ucf.edu/courses/cop4610L/spr2005

School of Computer Science
University of Central Florida
Dynamic Content in PHP

- Of all the strengths PHP exhibits as a server-side scripting language, perhaps its greatest strength lies in its ability to dynamically change XHTML output based on user input.

- In this final section of notes, we’ll build on the examples we’ve constructed in the previous two sets of notes by combining form.html and form.php into one dynamic PHP document named dynamicForm2.php.

- We’ll add error checking to the user input fields and inform the user of invalid entries on the form itself, rather than on an error page. If an error exists, the script maintains the previously submitted values in each form element.

- Finally, after the form has been successfully completed, we’ll store the input from the user in a MySQL database.
Basically, the same registration form that was used in a previous example.
Hi Mark. Thank you for completing the survey. You have been added to the Pro Cycling mailing list.

The following information has been saved in our database:

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark Llewellyn</td>
<td><a href="mailto:markl@cs.ucf.edu">markl@cs.ucf.edu</a></td>
<td>(407)823-2790</td>
<td>Windows XP</td>
</tr>
</tbody>
</table>

Click here to view entire database.
Screen the user sees after clicking to see the entire database.

<table>
<thead>
<tr>
<th>ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>E-mail Address</th>
<th>Phone Number</th>
<th>Magazine</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Schumacher</td>
<td>Ralf</td>
<td><a href="mailto:ralf@toyota.com">ralf@toyota.com</a></td>
<td>(222)222-222</td>
<td>Cycling Weekly</td>
<td>Other</td>
</tr>
<tr>
<td>11</td>
<td>Thurau</td>
<td>Didi</td>
<td>agirl.de</td>
<td>(456)123-334</td>
<td>Mirror du Cyclisme</td>
<td>Windows 98</td>
</tr>
<tr>
<td>10</td>
<td>Jolie</td>
<td>Angelina</td>
<td>hollywood.com</td>
<td>(123)456-789</td>
<td>Velo-News</td>
<td>Windows XP</td>
</tr>
<tr>
<td>9</td>
<td>Schumacher</td>
<td>Michael</td>
<td><a href="mailto:michael@ferrari.it">michael@ferrari.it</a></td>
<td>(407)823-279</td>
<td>RadSport</td>
<td>Windows 2000</td>
</tr>
<tr>
<td>13</td>
<td>Campbell</td>
<td>Kristy</td>
<td>im_not_sure</td>
<td>(345)987-654</td>
<td>Cycle Sport</td>
<td>Windows 2000</td>
</tr>
<tr>
<td>14</td>
<td>Einstein</td>
<td>Albert</td>
<td>relative</td>
<td>(111)222-333</td>
<td>Cycling Weekly</td>
<td>Other</td>
</tr>
<tr>
<td>16</td>
<td>Llewellyn</td>
<td>Mark</td>
<td><a href="mailto:markl@cs.ucf.edu">markl@cs.ucf.edu</a></td>
<td>(407)823-279</td>
<td>Pro Cycling</td>
<td>Windows XP</td>
</tr>
</tbody>
</table>
Dynamic nature of the PHP form is illustrated when the user fails to enter proper information into the form. In this case, the user forgot to enter their first name. Error checking is in place on each user input location and the page is dynamically updated to reflect the error processing and correction capabilities. The database will not be updated until the user has correctly filled in all required fields.
Screen shot from MySQL of the contacts relation after the inclusion of several users. Note that the values in the table are the same as those returned to the PHP document in the previous slide.
<?php
extract ( $_POST );
$iserror = false;
// array of magazine titles
$maglist = array( "Velo-News",
    "Cycling Weekly",
    "Pro Cycling",
    "Cycle Sport",
    "RadSport",
    "Mirror du Cyclisme" );
// array of possible operating systems
$systemlist = array( "Windows XP",
    "Windows 2000",
    "Windows 98",
    "Linux",
    "Other" );
// array of name and alt values for the text input fields
$inputlist = array( "fname" => "First Name",
                    "lname" => "Last Name",
                    "email" => "Email",
                    "phone" => "Phone" );
if ( isset ($submit) ) {
    if ( $fname == "" ) {
        $formerrors[ "fnameerror" ] = true;
        $iserror = true;
    }
}
if ( $lname == "" ) {
    $formerrors[ "lnameerror" ] = true;
    $iserror = true;
}
if ( $email == "" ) {
    $formerrors[ "emailerror" ] = true;
    $iserror = true;
}
if ( !ereg( "^[0-9]{3}[0-9]{3}-[0-9]{4}$", $phone ) ) {
    $formerrors[ "phoneerror" ] = true;
    $iserror = true;
}
if ( !$iserror ) {
    // build INSERT query
    $query = "INSERT INTO contacts " .
             "( LastName, FirstName, Email, Phone, Magazine, OS ) " .
             "VALUES ( '$lname', '$fname', '$email', "$phone", '$mag', '$os' )";
// Connect to MySQL
if ( !( $database = mysql_connect( "localhost", "root", "root" ) ) )
die( "Could not connect to database" );

// open MailingList database
if ( !mysql_select_db( "MailingList", $database ) )
die( "Could not open MailingList database" );

// execute query in MailingList database
if ( !( $result = mysql_query( $query, $database ) ) ) {
    print( "Could not execute query! <br />" );
die( mysql_error() );
}
print( "<p>Hi
    <span style = 'color: blue'> <strong>$fname</strong></span>.
    Thank you for completing the survey.<br />
    You have been added to the <span style = 'color: blue'>
    <strong>$mag</strong></span> mailing list.       </p>
    <strong>The following information has been saved in our database:</strong><br />
    <table border = '0' cellpadding = '0' cellspacing = '10'>
    <tr><td bgcolor = '#ffffaa'>Name </td>
    <td bgcolor = '#ffffbb'>Email</td>
    <td bgcolor = '#ffffcc'>Phone</td>
    <td bgcolor = '#ffffdd'>OS</td></tr><tr>
    <td bgcolor = '#fffffaa'>$name</td>
    <td bgcolor = '#fffffbb'>$email</td>
    <td bgcolor = '#fffffcc'>$phone</td>
    <td bgcolor = '#fffffdd'>$os</td>
    </tr>
    </table>
<!-- print each form field's value -->
<tr><td>$fname $lname</td><td>$email</td><td>$phone</td><td>$os</td></tr></table><br /><br /><br />
<div style = 'font-size : 10pt; text-align: center'>
<div style = 'font-size : 18pt'>
<a href = 'formDatabase2.php'>
Click here to view entire database.</a>
</div></div>

</div></body></html>
<img src='images/downloads.gif' alt='Publications' /><br />
<span style='color: blue'>Which magazine would you like information about?</span><br />
<select name='mag'>
</select>

The $$variable notation specifies variable variables. PHP permits the use of variable variables to allow developers to reference variables dynamically. The expression $$variable could also be written as ${$variable} for added clarity.
Which operating system are you currently using?

<!-- create five radio buttons -->

$counter = 0;

foreach ( $systemlist as $currsystem ) {
    print("<input type = 'radio' name = 'os' value = '$currsystem'" );
    if ( $currsystem == $os ) print("checked = 'checked'");
    if ( iserror && $counter == 0 ) print("checked = 'checked'");

    print(" />$currsystem" );

    if ( $counter == 2 ) print("<br />");
    $counter++;
}

print("<!-- create a submit button -->
<br />
<input type = 'submit' name = 'submit' value = 'Register' />
</form></body></html>" );
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!– formDatabase2.php
<!-- Program to query a database and send results to the client. -->

<html xmlns = "http://www.w3.org/1999/xhtml">
<head>   <title>Database Search Results</title>   </head>
<body style = "font-family: arial, sans-serif"
style = "background-color: #F0E68C" background=image1.jpg>

<?php
extract( $_POST );
// build SELECT query
$query = "SELECT * FROM contacts";
   // Connect to MySQL
if ( !( $database = mysqli_connect( "localhost",  "root", "root", MailingList ) ) )
        die( "Could not connect to database" );
   // query MailingList database
if ( !( $result = mysqli_query( $database, $query ) ) ) {
        print( "Could not execute query! <br />");
        die( mysqli_error() );
}
<h3 style = "color: blue">
Mailing List Contacts</h3>
<table border = "1" cellpadding = "3" cellspacing = "2" style = "background-color: #ADD8E6">
<tr>
<td>ID</td>
<td>Last Name</td>
<td>First Name</td>
<td>E-mail Address</td>
<td>Phone Number</td>
<td>Magazine</td>
<td>Operating System</td>
</tr>
<?php
// fetch each record in result set
for ( $counter = 0; $row = mysqli_fetch_row( $result ); $counter++ ){
    // build table to display results
    print( "<tr>" );
    foreach ( $row as $key => $value )
        print( "<td>$value</td>" );
    print( "</tr>" );
}
mysqli_close( $database );
?>
</table>
</body>
</html>
Schema of the MailingList database table contacts required for the PHP database example to work.